Indexing multispectral images for content-based retrieval *

Julio Barros, James French, Worthy Martin

Computer Science Dept.
University of Virginia
Charlottesville, Va 22903

Patrick Kelly, James M. White

Los Alamos National Laboratory
Los Alamos, NM 87545

ABSTRACT

This paper discusses our view of image databases, content-based retrieval, and our experiences with an experimental system. We present a methodology in which efficient representation and indexing are the basis for retrieval of images by content as well as associated external information. In the example system images are indexed and accessed based on properties of the individual regions in the image. Regions in each image are indexed by their spectral characteristics, as well as by their shape descriptors and position information. The goal of the system is to reduce the number of images that need to be inspected by a user by quickly excluding substantial parts of the database. The system avoids exhaustive searching through the image database when a query is submitted.

1 INTRODUCTION

Recent advances in image acquisition, storage, processing, and display capabilities have made the use of digital imagery data affordable and prevalent in many situations. These image collections grow quickly and soon become difficult to manage. Without effective organization it becomes tedious and time consuming to find necessary images and the full value of the images and the collection cannot be exploited. This paper discusses issues with digital imagery databases and our approach and experiences with content-based retrieval mechanisms. By an *image database* we mean a collection of images that has been organized for convenient and efficient access. The emphasis is on convenient and efficient access. Many applications refer to any collection of images as a database. These systems typically exhaustively search the collection for relevant images. However, as these collections grow, exhaustive searches become less feasible. *Content-based retrieval* is the searching of image databases based on the intrinsic properties of the images. Ideally we would like the semantic meaning of an image to be the intrinsic property used. However, automating this is not currently feasible. Part of the purpose of this research is to explore what techniques are useful and needed in bridging the gap between what is desirable and what is possible.

Currently, many image collections are simply archived in traditional file systems. In these collections, an image's name and location convey information about the content. In more advance archives, information about,

*To appear in the proceedings of the 23rd AIPR Workshop on Image and Information Systems, Washington DC, Oct 1994.

or that describes, an image is manually extracted and maintained in an auxiliary file or with the image files themselves. In these systems images are selected by browsing the file system or by searching the metadata file.

The main reason for the popularity of these approaches is that these systems are simple to set up and use. Individuals are not restricted by established conventions regarding data content, structure or usage. Also, without complicated data structures, data access is a straightforward process. Another reason these approaches are used is that there is little, if any, additional software to purchase or maintain. Image archives can be maintained with only the software already available on many computer systems. Without a major initial investment of time and money these systems appear inexpensive.

However, these approaches have several drawbacks. In these systems the query facilities are comparatively primitive. It is neither convenient nor efficient to find relevant images. Searches are typically performed sequentially on a single list of file names or metadata. This limits the complexity, power, and efficiency of the expressible queries. These approaches do not scale well. As the image collection grows so does the number and size of files. Classifying the images, managing the files and ensuring that the metadata is accurately extracted is time consuming and error prone. In many applications, it is important to know when and how a piece of metadata was generated. Without a mechanism to manage the metadata a user cannot be certain of the value of the information.

The ad hoc organization does not allow easy sharing of data. Without established conventions and a clear understanding of the available information it is difficult to determine what may be of value to others and how it may be extracted from the collection. If the information cannot be conveniently shared between people and application programs it cannot be fully used. Consequently, time and money are wasted recalculating data or re-retrieving images. With all the drawbacks of these simple archives it becomes apparent that a tremendous amount of information is lost and time wasted. Using a database management system (DBMS) to help alleviate these problems becomes increasingly attractive.

In other current image databases, the metadata includes external information such as imaging parameters and a textual description of the subject matter. This data is organized and maintained by a DBMS. Queries processed by these systems operate on the attached information. Therefore, the more complete the description, the more useful it will be during searches. Retrieval in these systems is very similar to information retrieval from text databases.

However, it is not feasible, using manual annotation methods, to describe a complex image completely and uniquely. Additionally, the textual descriptions are not only time consuming to enter but are inadequate to handle changing retrieval needs. Ideally, we would like an image database with automated "semantic retrieval" capability. This would allow a user to ask questions on the *meaning* of an image with out having previously provided such information. However, it is not possible to automatically retrieve images based purely on their semantics since automating this requires computer vision, analysis, and interpretation techniques that are far beyond current capability.

Current image processing techniques process images to extract mid-level *features* from images. These features can serve, in a limited way, as metadata in an image database. Feature-based systems[5,6,17,19,23] are currently being developed and used. In these systems, the feature information is typically numeric and can be treated as a vector. For efficiency reasons, the vectors are organized in a multi-dimensional data structure. The similarity between two images is measured by the Euclidean or weighted Euclidean distance between their feature vectors. Queries are answered by retrieving the k images that are most similar to a query image or by retrieving all images within some distance of a query image.

Though these powerful new systems show great promise in many applications they have several drawbacks. These systems do not adequately address the notion of similarity. It is not clear exactly how distance in feature space translates to similarity in the users opinion. Admittedly, this is a difficult problem. However, if a system is attempting to retrieve perceptually similar images then the feature and similarity metric should have some human

perception rationale. For example, shape descriptors have varying correspondence to human shape perception.[22] Also, the human visual system has different sensitivity in each of the red, green, and blue (RGB) bands. If a system is using a weighted Euclidean distance in RGB space, perhaps the weights should mimic the response of the human visual system. Other features and metrics should be used where appropriate. Users without an intuitive notion of the similarity metric being used are likely to be surprised by the results.

Feature-based systems tend to be application specific. In general they do not provide guidance or allow flexibility in feature selection. The features used ultimately determine the types queries that can be expressed. Each application will have different retrieval needs and thus will require different feature sets. The use of a fixed feature set with little explanation or justification limits the usefulness of these systems.

These systems are not flexible enough to change features as retrieval needs change. The database administrator decides on the schema before the database is populated and put in use. Changing the schema, adding or removing a feature, is a significant undertaking. This is a nuisance in traditional databases and limits the utility of feature-based databases. It would be very convenient to be able to add features as new ones are discovered, developed and deemed useful. Consequently, we must extend feature-based systems and develop true image databases with content-based retrieval methods and advanced data management facilities. The rest of this paper discusses our view of image databases, content-based retrieval, and our experiences with an experimental system.

2 IMAGE DATABASES

Image databases, as described above, are different than traditional databases. In traditional databases the basic unit of information is a well defined value. This value has a clear and precise semantic meaning and can be compared to other related values. In images the basic level of information is the pixel. Pixels are the value of a measurement by a sensor of a property of a region in space. Although it is possible for the value of a pixel to have a clear quantitative meaning, normally it, in addition to having been perturbed by sensor noise, is ambiguous as to the specific object it represents. Consequently, the meaning of a pixel is at a level far below the semantic meaning desirable for image interpretation. For this reason, it is not sufficient to maintain a database simply on the values of pixels. The ability to process exact value queries is usually not interesting. It is necessary to be able to pose and process queries at a higher-level. Image processing and computer vision techniques attempt to convert raw pixel data into a more meaningful representation. However, it is not clear that it is possible to use this intermediate representation to answer general higher-level queries. Addressing this issue is a key part of our research.

Schema management and data access patterns in image databases are also different from traditional databases. In traditional systems considerable effort is expended designing the best schema possible before data is ever entered into the system. In traditional databases, once a schema is established it is rarely changed. However, an image database may need to be able to support schema changes during operation. These changes would be required as new image processing routines are developed and adopted or as retrieval needs change. An image database needs to handle insertions, queries, and possibly deletions, efficiently. However, feature data rarely needs to be updated. Once a particular feature is calculated its value does not change unless the feature extraction process is modified or replaced. However, this is now a different feature and not an update of the old feature. For example, in a database of satellite imagery, users would be continually adding images. As new feature extraction routines are developed they can be used added to the system. However, since the images don't change, the values of the features don't change unless the feature extraction routine is modified. We can see that value modification will be rare and that schema modification might be frequent when compared to traditional database systems.

When setting up an image database, it is necessary to identify which features can or should be used in that particular domain. Traditionally, this has been performed by a system designer with significant domain experience. The selection of features to index is critical because it defines the types of queries that can be posed

and then processed efficiently. It is unlikely that this aspect of image database design can be fully automated. Addressing this issue is another part of our research.

In our view an image database can conceptually be broken down into a database management system and an *insertion module* and a *retrieval module* (see Figure 1). The insertion module assists with the selection and maintainance of feature data while the retrieval module assists with specifying and processing user queries.

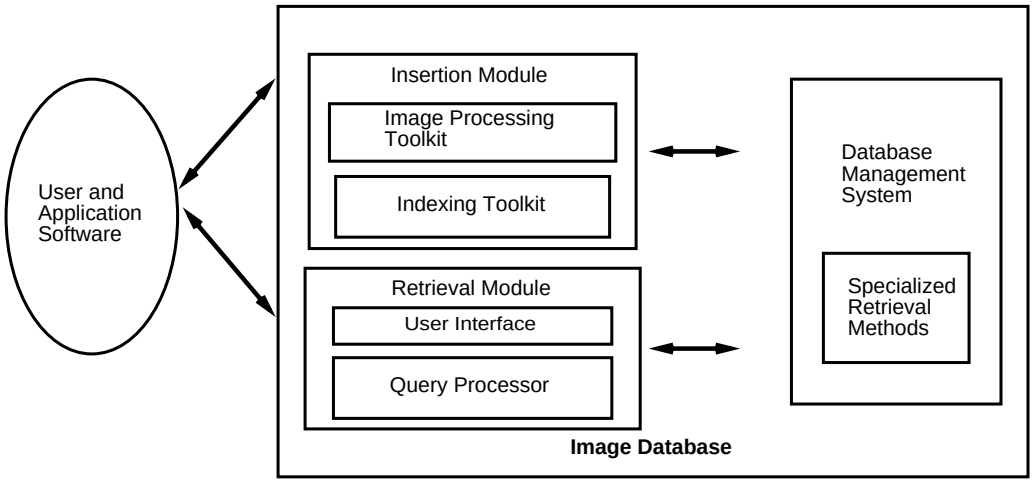

Figure 1: The components of an image database.

The insertion module includes an image processing toolkit and an indexing mechanism toolkit. This module assists a designer in defining and maintaining the features appropriate to the application. Images can be processesed as they are entered into the system. General purpose image processing toolkits include operations that produce images and operations that produce feature data. Although the former are an essential part of the toolkit, it is not any easier to index their output than it was to index the original image. For this reason, we concentrate on operations that produce feature data. Typically, these operations accept the original image, a processed image, or other feature data. The results of the processing are stored and organized in an index similar to that in a traditional database.

In the insertion module, basic image processing routines can be combined to form more sophisticated operations. Each combination needs to have an input and at least one output with a suitable indexing mechanism. Single image processing operations do not need to be distinguished logically from combinations of operations. The insertion module maintains information on how to create the feature data for each new image. This information is necessary both to process new images and to provide a record of the procedures and parameters used in extracting the feature data. In many applications the feature data would be useless with out a clear knowledge of how it was generated.

The indexing toolkit contains operations that organize feature data to allow rapid access. This toolkit does not restrict the use of textual descriptions or annotations, and would be used in addition to current methods. Different indexing mechanisms are needed to handle the various types of feature data. Feature data that is composed of single values can be maintained by any of several common data structures. Multi-value feature data would require more complex indexing mechanisms. Some can be stored as records while others need to be broken up into multiple tables or processed further. It is also necessary for the database system to efficiently handle the modification of tables. This will be needed when new feature data and indexes are added to a previously existing database.

The operations in an image processing toolkit need to be examined and matched with an appropriate indexing

mechanism. Not all types of feature data will have a satisfactory indexing mechanism. For an insertion module to be generally useful it needs to provide many different types of feature data and corresponding indexing mechanisms. Each of these operations and indexing mechanisms must have an explanation of its performance characteristics. Since features need to be matched to retrieval needs, this information is critical to the designers and users of the image database. In order capitalize on the strengths and minimize weaknesses of the features, several could be used for each application.

We are interested in feature indexes purely for efficiency reasons. Indexing does not provide a gain in computational power. Indexing does not provide more information with which to answer queries. However, if done well, it does provide a significant increase in retrieval performance. That is, an index is an organization of the feature information that allows efficient access. Indexing every image is potentially computationally expensive. However, users may be willing to incur the high *initial* computational and maintainance cost if it is offset by a gain in lower *operational* cost. The index must be useful for multiple queries for it to actually be beneficial.

The retrieval module allows a user to express a query and efficiently retrieves the requested images. This is likely to be an iterative process, therefore an interactive retrieval module may be particularly useful. A command language is the most direct query language but requires the user to understand what is indexed and how to coerce the index to retrieve what is required. Query by pictorial or image example (QPE)[1] is another query method. With a QPE system a user provides an image example as the query image and specifies which aspects are relevant. For example, a user may provide an image, select a region, and then specify texture or color as the relevant characteristic. Developing efficient mechanisms for expressing queries is an important research topic that needs to be examined further. Once a user expresses a query, the query processor uses the indexes created by the insertion module to retrieve the images of interest efficiently.

The effectiveness of an image database system depends on the existence of useful general image features, the ability to formulate a query, and the ability to efficiently process a query. An ideal general image feature would have some or all of the following characteristics.

- The feature should have intuitive meaning to the user. That is, there should be a connection or mapping between the user's need and the information a feature can provide. A clear relation between feature and user need is necessary for the effectiveness of the retrievals and the ease of specifying queries. Determining the relation between high-level user needs and a middle-level feature indexes is the thrust of our research.
- The feature should be able to discriminate between relevant and non-relevant images. In the absence of perfect discrimination ability these features will not necessarily return exactly the set of desired images, but rather, will quickly exclude large parts of the database. This will reduce the number of images a user need inspect.
- For efficient retrieval the feature should be matched to an appropriate indexing mechanism. That is, there should be a method to quickly extract a subset of images based on the feature. Powerful features with no known indexing method can be used in the last stages of the query process. Once a relatively small set of images has been retrieved from the database it may be practical to process the images individually. As an example, take a user that frequently needs to search a collection of images based on the size and shape of objects in the images. The shape matching procedure may be computationally expensive and there may not be an appropriate indexing mechanism available. However, it is straight forward to index by size. So, in this case it makes sense use the size index first since it is quick and will reduce the number of images that need to be processed by the shape matching procedure.
- The feature should be computationally inexpensive relative to the reduction in operational costs. The maintainance of feature data will incur additional overhead. This cost can only be justified if the feature data is useful in multiple queries. The cost of an inexpensive feature can be amortized over fewer queries. For example, suppose that in the previous scenario size was rarely used in queries. There is a computational cost, however small, associated with extracting the size information and maintaining the index. For the

size feature the cost can be recovered if it is useful in a small number of queries. However, more expensive features will need to be more general and applicable before it is advantageous to create appropriate indexes. In other words, a balance must be attained between the cost and utility of each index.

The next section of this paper describes an example application, some experimental features, an indexing mechanism and a prototype interface.

3 AN EXAMPLE

As our initial application we have chosen the support of analysis of remotely-sensed multispectral imagery such as Landsat Thematic Mapper (TM) data. Landsat TM data is multispectral in that it is sensed in seven spectral bands simultaneously. This domain was chosen because typical volumes of this data are enormous. A Landsat TM scene is about 10 megabytes per band or 70 megabytes for all 7 bands. The total number of scenes that could possibly be acquired is about 600 per day. However, it is difficult to estimate the current actual number acquired. In the near future, this type of data will be collected in such quantities that only a small portion will ever be viewed by a human interpreter. Although our system is described in terms of a remotely-sensed data application, it can easily be extended to other problem domains.

Content-based retrieval methods can work with both images as a whole or with parts of images. Queries can then be posed at either level. The CANDID[13,14] project suggests that a global signature can be used to retrieve whole images from a database. Our example in this paper, focuses on retrieval at the component region level. Regions are homogeneous areas segmented from the image. Queries are processed on feature data extracted from the regions and the user gets a list of regions as the answer to the query.

Each image is segmented into regions. Each region is then indexed by its spectral characteristics, shape descriptors, and position information. Retrieval mechanisms include searches on single scalars and distributions of values and, for efficiency reasons, avoid exhaustively searching the entire image database. We use Khoros[20] as our image processing toolkit and custom software for indexing and retrieval.

3.1 SEGMENTATION

Segmentation is an important and early part of the classification and analysis process. It is also the basis of the indexing process we describe. Several authors[9,18,21] provide an overview of common segmentation techniques. However, difficulties arise with multispectral data because of the large memory requirements, lack of available multidimensional algorithms, mixed pixels (pixels representing areas consisting of more than one distinct land-use type) and similar spectral signatures for logically different classes.

Many segmentation procedures attempt to separate an image into foreground and background by finding regions or edges. Threshold-based approaches accomplish this separation by finding an optimum level in the histogram of the pixel values. All pixels above this level are labeled foreground and all pixels below are labeled background. How an optimum level is defined depends on the application. This approach does not work well for remotely-sensed data because the histograms, as seen in Figure 2, are typically concentrated in a single range rather than being distributed over several distinct ranges. This is caused by the spectral values of the classes overlapping significantly in any given dimension. Another common approach is to attempt to find the *edges* between regions. Edges are areas of relative discontinuity (as opposed to regions which are areas of relative continuity) in the pixel values. Although good edge-finding algorithms exist,[11] they can be sensitive to noise and produce false edges and incomplete edges. Additionally, If the bands of a multi-band image are examined individually, with either approach, it is difficult to combine the results into a final answer.[2,8]

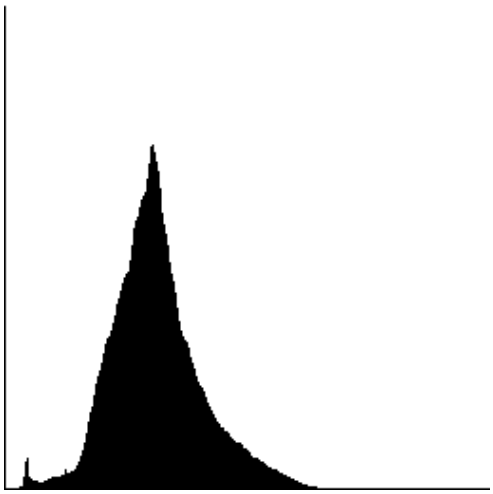

Figure 2: Histogram for band 4 of an image.

Clustering is an approach to segmentation and data reduction that can consider all available dimensions.[4,15] However, many clustering algorithms require that we know the number of clusters *a priori*. Ideally we would like one cluster per logical class but we may not know how many classes are actually present in the image. Additionally, for accurate representation, multi-modal classes require multiple clusters. Confusion is also introduced by pixels that represent more than one logical class. In these mixed pixels, the spectral values are a mixture of the included classes. Furthermore, clustering algorithms attempt to minimize variance across all clusters, while logical classes may actually have extremely different variances. Consequently, the clustering algorithm may be at odds with our semantic understanding.

The observation that two pixels with a given spectral distance to each other are more likely to be from the same class if they are also close spatially led us to our approach. Like other techniques,[3] ours uses both spectral and spatial information. We proceed in two stages. The first stage uses the k-means clustering algorithm to cluster pixels spectrally into a large number of clusters. The second stage reduces the number of clusters by using both spectral distance as well as spatial (coocurrence) information to combine clusters. *Spectral information* is the distance between clusters in the multi-dimensional spectral space and the *spatial information* is captured in the coocurrence count. Coocurrence matrices of textures have been used in image analysis for several years.[7] For our purposes, the cooccurrence count is the number of times that pixels assigned to two given clusters are adjacent in the image. This is calculated by keeping count of the cluster assignments for the eight neighbors around each pixel.

The spectral and spatial information is used to calculate a score for each pair of clusters. The score attempts to find two clusters that are spectrally similar and occur together frequently in the image. In the formula below, $coocurrence_{ij}$ is the number of times cluster i appears adjacent to cluster j, $number_i$ is the number of pixels in cluster i, and $distance_{ij}$ is the effective merging radius[12] between cluster i and cluster j. The effective merging radius, like the mahalanobis distance, is not a Euclidean distance and takes into account the region variances.

$$score_{ij} = \frac{coocurrence_{ij}}{number_i} * \frac{coocurrence_{ji}}{number_j} * \frac{1}{distance_{ij}}$$

During the merging step this score is calculated for all pairs of clusters. The two clusters with the highest score are combined, and the process is repeated. Note that we only need to recompute the scores for pairs that had previously involved either of the selected clusters. Clusters are combined in this manner until a threshold score

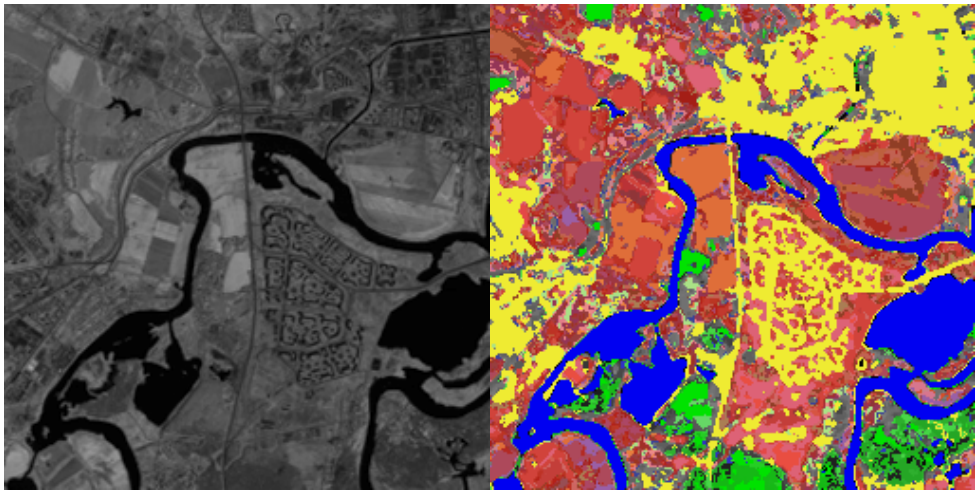

Figure 3: Subsection of band 4 of an image. Original gray level and segmented with pseudo coloring to highlight label differences.

---

is reached, some number of merges are performed, some number of clusters are left, or all clusters are assigned to predetermined *distinguished* classes. An example is shown in Figure 3, a subsection of band 4 of an image of Moscow. The original gray level image is shown along side the segmented image. In an attempt to highlight the different classes in the segmented image, regions are given an arbitrary color (gray level).

Although we have observed good empirical results and have done some test work on synthetic images, more work needs to be done to measure the performance of this method and to find a procedure to automatically decide when to stop merging the clusters. We are currently analyzing the performance of this method by using real Landsat data that has been carefully analyzed and classified by hand.

3.2 THE INDEXES

Once we have an acceptable segmentation, features for each of the regions are extracted and maintained in the database. The features include area, center, extent of bounding rectangle, orientation, eccentricity, invariant moments,[10] and spectral means and covariance. The area is calculated to allow queries on the size of a region. The center and bounding rectangle allow position and spatial relation queries. The orientation, eccentricity, and invariant moments allow queries on the shape and pose of the region. The spectral information allows queries on the spectral signature of the regions.

For many of the features, such as size and position, a traditional index of key value pairs is built and used for efficient retrieval. Other features such as spectral means and variances, are not single scalars and thus need to be managed differently. Searching for all regions with means *exactly* equal to a particular value is typically not useful. A more interesting search is one for all regions *similar* to a particular value. However, we are left with the problem of specifying what we mean by similar. In other image databases[17] users express the notion of similarity by specifying a distance or range in the query. This requires the user to know and understand the variability of the data in order to select a good range. In our application, the spectral values of the pixels in a region are viewed as a normal distribution. Instead of querying for regions with a particular mean we query for regions with means that *could*, with a given confidence level, be a particular value. This approach is similar but not the same as a k-nearest neighbors search. In the k-nearest neighbors search we specify, k, the number of

regions to be returned in the result set. In our approach, we specify the level of similarity (confidence) between the query point and each of the regions in the resulting set.

The spectral information for each region is calculated from the values of the pixels in the region. Either the actual values from the raw data or the means from the first-stage clusters can be used. Even though two regions may ultimately belong to the same class, their pixels will have different original values, or will have come from different first-stage clusters, so they will have different means and covariances. The matching process then becomes the problem of selecting the distributions in the database that could reasonably have means equal to the specified query point. This is done with the two tailed statistical hypothesis-testing procedure[16] below. Note that this test essentially calculates the number of standard deviations allowed between the query point and the sample mean. In this manner the region spectral variance and size specifies how close to the region mean a query point has to be in order to be judged similar to that particular region. We want to test the hypothesis that the mean is equal to the query point against the alternate hypothesis that it is not equal. α is the confidence level between 0 and 1. n_i is the number of samples in region i. μ is the query point. $\bar{X}_i$ is the region or sample mean. The region variance s_i^2 is used as the population variance σ^2 since the number of samples is large.

$$Z_i = \frac{\bar{X}_i - \mu}{s_i/\sqrt{n_i}}$$

We can calculate the critical value of $z_{\alpha/2}$ from a standard table and accept the hypothesis if $z_{\alpha/2} < Z_i < z_{1-\alpha/2}$.

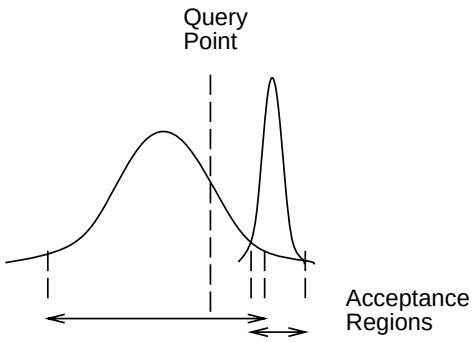

Figure 4: Measuring similarity to a query point based on region size and variance.

For example, Figure 4 shows a query point and the spectral information for two candidate regions. Though the two region means are approximately the same distance from the query point one region is judged similar while the other is not.

As written above, the test would be applied consecutively to each distribution in the database. Those passing the test are kept as possible matches to the query while the others are rejected. By rewriting the equation and processing the query in two steps, we can avoid exhaustively examining all the distributions in the database. First we rewrite the expression as shown below.

$$\mu + \frac{s_i z_{\alpha/2}}{\sqrt{n_i}} < \bar{X}_i < \mu + \frac{s_i z_{1-\alpha/2}}{\sqrt{n_i}}$$

This tells us the range that $\bar{X}_i$ must lie in to pass the test. Then, by using the largest s and the smallest n in the database we can calculate a range that all $\bar{X}$ must lie in to possibly pass the test.

$$\mu + \frac{s_{max} z_{\alpha/2}}{\sqrt{n_{min}}} < \bar{X} < \mu + \frac{s_{max} z_{1-\alpha/2}}{\sqrt{n_{min}}}$$

This range contains the means of all the distributions that could possibly match the query point as well as some that do not.

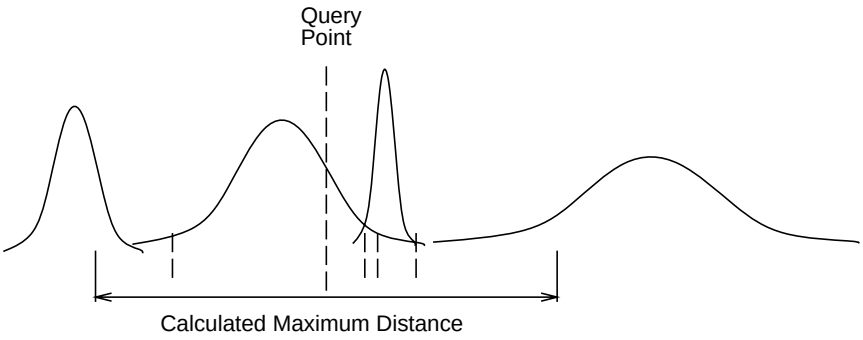

Figure 5: The range of possible matches.

Querying the index is now a two step process. First we calculate range just described and use an efficient range search to retrieve all the regions in this range. We then take all the regions in this subset and examine them individually using the individual region variance and size. The regions that pass the individual tests are then returned as the answer set. For example, of the four regions in Figure 5 two would be extracted by the range search and one will be returned as the answer set.

Each index specified in the query is searched independently. The intermediate-result sets from each index are intersected, and the final result is returned to the user. This approach can be extended to allow the user to specify a query range and a confidence level.

3.3 PROTOTYPE QUERY INTERFACE

In order to experiment with the indexes we built a simple interface. The interface allows users to pose several types of interesting queries. The search specification can be expressed as either a fully or partially specified region. It is currently possible to query by region size, position, eccentricity, angle, and spectral information. The indexes can be used in any combination, and the results appear in a text list. If an item from this list is selected, the corresponding region is extracted from the second-stage clustered image and displayed. By manipulating the color map, the region in question can be made to stand out against other regions.

For example, it is possible to request regions with spectral signatures similar to clouds. The system then lists all regions with a similar spectral characteristic. The user can then visually inspect the listed regions. The query can also be restricted to regions of a certain size or position. Although not possible with the current interface, information is available in the indexes to answer certain queries about images as a whole. For instance, it should be possible to process queries such as "show images that contain small cloud like regions" as well as other queries on the statistics of regions in images.

4 CONCLUSION AND FUTURE RESEARCH

The current text-based approach and low-level of automation in image databases results in slow retrievals and is expensive in human resources. The development of image databases with content-based retrieval will be useful in many different domains. However, we believe that high-level semantic retrieval is unattainable in the near future. Consequently, it may be more productive to focus on mechanisms for middle-level content-based retrieval.

The effectiveness of an image database system depends on the existence of useful general image features, the ability to formulate a query, and the ability to efficiently process a query. The maintainance of feature data will incur additional overhead that can be be justified only if the feature data is useful over multiple queries. A clear relation between features used and user need is necessary for the effectiveness of the retrievals and the ease of specifying queries. Determining the relation between high-level user needs and a middle-level feature indexes is the thrust of our research.

We claim that a balance must be struck between the degree of effectiveness in satisfying general queries and the degree of efficiency in retrieval through image feature indexes. For this reason the goal of a system should not be to return specifically the images that satisfy the query, but rather to retrieve a small (relative to the size of the total image collection) set of images that cover the query. In databases terms, we want the system to sacrifice precision for efficiency while maintaining maximum recall. The intent here is that it is more practical to browse or apply more computationally expensive procedures to the greatly reduced set of images.

In this paper we present a method to index and retrieve image regions based on spectral distributions and a system for experimenting with content-based retrieval mechanisms. As an example we presented a feature that is technical but has a semantic meaning to analysts. We have shown how the notion of similarity can be expressed, how the user can control it, how it can be measured, and how it can be used for efficient retrieval. We are still working on these indexing mechanisms and future work will address questions as to its utility, effectiveness, and efficiency.

5 ACKNOWLEDGMENTS

Portions of this work were were performed under a U.S. Government contract (W-7405-ENG-36) by Los Alamos National Laboratory, which is operated by the University of California for the U.S. Department of Energy. This work was also supported by an NSF Graduate Engineering Education fellowship.

6 REFERENCES

[1] N. S. Chang and K. S. Fu, "Picture query languages for pictorial data-base systems," *Computer*, November 1981.

[2] Chen-Chau Chu and J. K. Aggarwal, "Image interpretation using multiple sensing modalities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):840–847, August 1992.

[3] Steven E. Franklin and Bradley A. Wilson, "Spatial and spectral classification of remote-sensing imagery," *Computers and Geosciences*, 17(8):1151–1172, 1991.

[4] Tung Fung and King chung Chan, "Spatial composition of spectral classes: A structural approach for image analysis of heterogeneous land-use and land-cover types," *Photogrammetric Engineering & Remote Sensing*, 60(2):173–180, February 1994.

[5] James E. Gary and Rajiv Mehrotra, "Shape similarity-based retrieval in image database systems," In Albert A. Jamberdino and Wayne Niblack, editors, *Image storage and retrieval systems*, pages 2–8, Bellingham, Washington, February 1992. SPIE.

[6] William I. Grosky and Zhaowei Jiang, "A hierarchical approach to feature indexing," In Albert A. Jamberdino and Wayne Niblack, editors, *Image storage and retrieval systems*, pages 9–20, Bellingham, Washington, February 1992. SPIE.

[7] J. F. Haddon and J. F. Boyce, "Co-occurrence matrices for image analysis," *Electronics and Communication Engineering Journal*, 5(2):71–83, April 1993.

[8] John F. Haddon and James F. Boyce, "Image segmentation by unifying region and boundary information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):929–948, October 1990.

[9] Robert M. Haralick and Linda G. Shapiro, "Survey: Image segmentation techniques," *CVGIP: Image Understanding*, 29:100–132, 1985.

[10] Ming-Kuei Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, IT-8:179–187, February 1962.

[11] A. I. Katsevich and A. G. Ramm, "Multidimensional algorithm for finding discontinuities of signals from noisy discrete data," Los Alamos National Lab. Unpublished.

[12] Patrick Kelly, "An algorithm for merging hyperellipsoidal clusters," Technical Report LA-UR-94-3306, Los Alamos National Laboratory, 1994.

[13] Patrick M. Kelly and T. Michael Cannon, "CANDID: Comparison algorithm for navigating digital image databases," In *Proceedings of the Seventh International Working Conference on Scientific Database Management*, Charlottesville, Virginia, September 1994.

[14] Patrick M. Kelly and T. Michael Cannon, "Experience with CANDID: Comparison algorithm for navigating digital image databases," In *Proceedings of the 23rd AIPR Workshop on Image and Information Systems: Applications and Opportunities*, Washington, D.C., October 1994.

[15] Patrick M. Kelly and James M. White, "Preprocessing remotely-sensed data for efficient analysis and classification," In *SPIE Vol. 1963 Applications of Artificial Intelligence 1993: Knowledge-Based Systems in Aerospace and Industry*, 1993.

[16] William Mendentall, Dennis D. Wackerly, and Richard L. Schearffer, *Mathematical Statistics with Applications*, PWS-KENT Publishing Company, Boston, Ma, 4 edition, 1990.

[17] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. "The QBIC project: Querying images by content using color, texture and shape," In Wayne Niblack, editor, *Storage and Retrieval for Image and Video Databases*, pages 173–187, Bellingham, Washington, February 1993. SPIE.

[18] Nikhil R. Pal and Sankar K. Pal, "A review on image segmentation techniques," *Pattern Recognition*, 26(9):1277–1294, 1993.

[19] A. Pentland, R. W. Picard, and S. Sclaroff, "Photobook: Tools for content-based manipulation of image databases," In Wayne Niblack and Ramesh C. Jain, editors, *Storage and Retrieval for Image and Video Databases II*, pages 34–47, Bellingham, Washington, February 1994. SPIE.

[20] John Rasure and Danielle Argiro, *Khoros User's Manual*, University of New Mexico, 1992.

[21] Todd R. Reed and J. M. Hans Du Buf, "A review of recent texture segmentation and feature extraction techniques," *CVGIP: Image Understanding*, 57(3):359–372, May 1993.

[22] Brian Scassellati, Sophoclis Alexopoulos, and Myron Flickner, "Retrieving images by 2d shape: A comparison of computation methods with human perceptual judgements," In Wayne Niblack and Ramesh C. Jain, editors, *Storage and Retrieval for Image and Video Databases II*, pages 2–14, Bellingham, Washington, February 1994. SPIE.

[23] N. Yazdani, M. Ozsoyoglu, and G. Ozsoyoglu, "A framework for feature-based indexing for spatial databases," In James C. French and Hans Hinterberger, editors, *7th Int. Working Conference on Scientific and Statistical Database Management*. IEEE Computer Society Press, September 1994.